Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления:

T. TT			
Kync //Hanaπi	игмы и конструк	IIIIII GOLIVAD MAA	Chammadaniada
турс мтаради	n widi n koncipyk	ции изыков про	

Отчет по лабораторной работе №3 «Функциональные возможности языка Python. Бот прогноза погоды»

Выполнил: Проверил:

студент группы ИУ5-32Б преподаватель каф. ИУ5

Фролов Илья

Подпись и дата: Подпись и дата:

Задание:

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Текст программы:

utils/tokens.py

```
import json
1
 2
 3
      def get_bot_token(): 1usage
           f = open('tokens.json')
           data = json.load(f)
6
           f.close()
           return str(data["tg-token"])
8
9
10
      def get_weather_token(): 1usage
11
           f = open('tokens.json')
12
           data = json.load(f)
13
14
           f.close()
           return str(data["weather-token"])
16
17
      def get_map_token(): 1usage
18
           f = open('tokens.json')
19
           data = json.load(f)
           f.close()
21
           return str(data["map-token"])
22
23
```

bot.py

```
return request["fact"]
  except:
    return "Error"
def parse_weather(data):
  # ТООО: сделать красивый вывод
  return str(data)
def get_cords_by_name(name):
  map_token = get_map_token()
  name = '+'.join(name.split())
  response_text = "https://geocode-maps.yandex.ru/1.x/?apikey=" + map_token + "&geocode=" +
name + "&format=json"
  request = requests.get(response_text).json()
  try:
    cords =
request["response"]["GeoObjectCollection"]["featureMember"][0]["GeoObject"]["Point"]["pos"]
    x, y = map(float, cords.split())
    return (x, y)
  except:
    return "Error"
def run():
  bot = telebot.TeleBot(get_bot_token())
  @bot.message_handler(commands=["start", "go"])
  def start(message):
    keyboard = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    button_geo = types.KeyboardButton(text="Отправить местоположение",
request_location=True)
    button_input = types.KeyboardButton(text="Ввести город")
    keyboard.add(button_geo)
    keyboard.add(button_input)
    bot.send_message(message.chat.id, "Поделитесь местоположением",
reply_markup=keyboard)
  @bot.message_handler(content_types=['text'])
  def weather_by_adress(message):
    adress = message.text
    if '/' in adress or adress == "Ввести город":
```

```
return
  try:
    lat, lon = get_cords_by_name(adress)
    data = get_weather((lat, lon))
    text = parse_weather(data)
    bot.send_message(message.chat.id, text)
  except:
    bot.send_message(message.chat.id, "Ошибка ввода локации!")
@bot.message_handler(content_types=['location'])
def weather_by_location(message):
  lat, lon = message.location.latitude, message.location.longitude
  data = get_weather((lat, lon))
  text = parse_weather(data)
  bot.send_message(message.chat.id, text)
bot.polling(none_stop=True, interval=0)
                                        main.py
                      1
                            from bot import run
                      2
                      3 🗅
                            ip __name__ == '__main__':
                                 run()
                     4
                      5
                                   Sensitive-данные:
nain.py
                                    bot.py

    tokens.json ×

                                                   tokens.py
Safe mode, limited functionality. Trust the project to access full IDE functionality.
 1
             "tg-token": "7085891070:AAGVHilje3zurBQ7b2FxcwpjzGbsqQUJJAA",
 2
             "weather-token": "69c05a34-7beb-4651-8b24-90b002394baa",
 3
            "map-token" : "accd9baa-ffb9-4c52-b61a-610f6d7e8c8a"
 5
```

Результаты исполнения:

weatherbot

бот



№ Написать сообщение...
 Отправить местоположение
 Ввести город